

University of Groningen

Hyperconnected Attribute Filters Based on k-Flat Zones

Ouzounis, Georgios K.; Wilkinson, Michael H.F.

Published in:
IEEE transactions on pattern analysis and machine intelligence

DOI:
[10.1109/TPAMI.2010.74](https://doi.org/10.1109/TPAMI.2010.74)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2011

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Ouzounis, G. K., & Wilkinson, M. H. F. (2011). Hyperconnected Attribute Filters Based on k-Flat Zones. *IEEE transactions on pattern analysis and machine intelligence*, 33(2), 224-239.
<https://doi.org/10.1109/TPAMI.2010.74>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Hyperconnected Attribute Filters Based on k -Flat Zones

Georgios K. Ouzounis and Michael H.F. Wilkinson, *Senior Member, IEEE*

Abstract—In this paper, we present a new method for attribute filtering, combining contrast and structural information. Using hyperconnectivity based on k -flat zones, we improve the ability of attribute filters to retain internal details in detected objects. Simultaneously, we improve the suppression of small, unwanted detail in the background. We extend the theory of attribute filters to hyperconnectivity and provide a fast algorithm to implement the new method. The new version is only marginally slower than the standard Max-Tree algorithm for connected attribute filters, and linear in the number of pixels or voxels. It is two orders of magnitude faster than anisotropic diffusion. The method is implemented in the form of a filtering rule suitable for handling both increasing (size) and nonincreasing (shape) attributes. We test this new framework on nonincreasing shape filters on both 2D images from astronomy, document processing, and microscopy, and 3D CT scans, and show increased robustness to noise while maintaining the advantages of previous methods.

Index Terms—Image enhancement, object detection, hyperconnectivity, connectivity, Max-Tree, document processing, anisotropic diffusion, attribute filter.



1 INTRODUCTION

CONNECTED morphological operators [1], [2], [3] are a set of powerful, robust, and computationally efficient tools that find use in image filtering [1], [4], [5], [6], [7], segmentation [8], [9], [10], and visualization [11]. Filters based on connected operators [12], [13], [14], [15], [16], [17], [18] can either remove a connected component or retain it unmodified, but cannot introduce new ones. This edge preserving property is highly desirable in many applications. If the filtering decision is based on some component attribute measure such as area or elongation, they are referred to as attribute filters [4]. An example is given in Fig. 1, which shows an image of the M81 galaxy together with many foreground stars. Star/galaxy classification is an important issue which can be handled well by mathematical morphology [19]. We aim one step further, namely, to remove the stars without distorting the internal detail of the galaxy, as was done in [20] using considerable manual intervention [21]. Even now, a more automatic method would be useful [22]. When using either an opening by a small euclidean disk or a median filter [22], many narrow structures in the galaxy are clearly distorted. By contrast, the area attribute filter (Fig. 1d) shows much better preservation of galaxy detail while removing most of the stellar signal. In this paper, we

will extend attribute filters using hyperconnectivity, to improve preservation of internal structure, combined with better suppression of background.

Like all connected operators, attribute filters rely on some notion of connectivity. This is usually modeled through the lattice-theoretic construct of *connectivity classes* [2], [12], [13], which allows for several generalizations [3], [7], [23], [24] either for enriching or further constraining image connections. Attribute filters based on standard connectivity or its derivatives are, in general, insensitive to contrast information. This can be limiting in applications handling texture-rich images when partitioning them to sets of flat zones [1] (connected regions of constant brightness). Any internal structure within the objects immediately fragments the corresponding image feature into a large number of flat zones. A possible solution lies in clustering flat zones, which is done *spatially* in second-generation connectivity. However, we might also want to cluster features in terms of gray scale, as in the case of the galaxy in Fig. 1a in which many gray levels clearly belong to the same object. Some attempts at adding tolerance to gray-level fluctuations within the flat zones framework have already been made.

Quasi-flat zones were previously proposed by [25] and used within mathematical morphology by Meyer [26] and Salembier et al. [27]. Salembier et al. [27] speak of “soft” binarization of gray-scale images, resulting in sets of quasi-flat zones. The latter are connected regions of variable intensity. From any given pixel in a quasi-flat zone, any other pixel of the same component can be reached through a path in which neighbors differ by no more than s , which is a measure of the *slope* allowed along a path. This prevents the oversegmentation issues of strict flat zones, but aggravates the leakage problem common to all connected filters [27]. This is shown in Fig. 2 where at $s = 1$; the entire image is just one quasi-flat zone because a path from any pixel to any other can be made in which gray levels between adjacent

• G.K. Ouzounis is with the Global Security and Crisis Management Unit, IPSC—Joint Research Centre—European Commission, T.P. 268, Via E. Fermi 2749, I-21027 Ispra (VA), Italy.
E-mail: georgios.ouzounis@jrc.ec.europa.eu.

• M.H.F. Wilkinson is with the Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, PO Box 407, 9700 AK Groningen, The Netherlands. E-mail: m.h.f.wilkinson@rug.nl.

Manuscript received 18 June 2008; revised 23 Mar. 2009; accepted 12 Jan. 2010; published online 2 Mar. 2010.

Recommended for acceptance by P. Maragos.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-06-0367.

Digital Object Identifier no. 10.1109/TPAMI.2010.74.

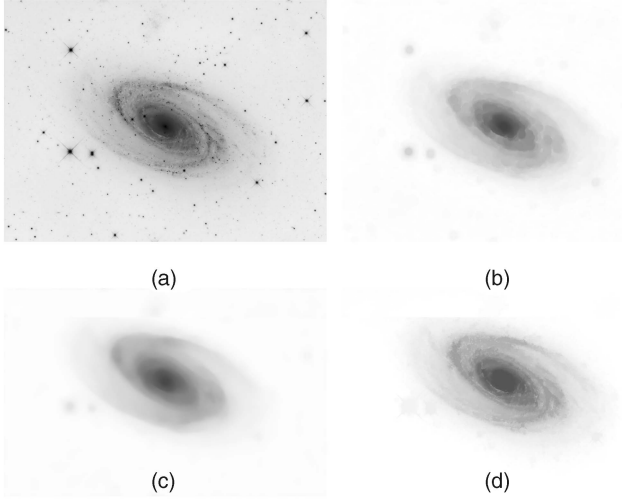


Fig. 1. Separating galaxies from stars. (a) Spiral galaxy M81, original image courtesy of Giovanni Benintende. (b) Stars suppressed by an opening with a disk of radius 8. (c) The result of a median filter applied on (a) using a disk of radius 16. (d) Area attribute filter with $2,000 \leq \text{area} \leq 240,000$. The latter shows the best preservation of edge detail and the best suppression of stellar signal.

pixels are no more than 1. Many attempts at resolving this have been proposed (for a review, see [28]), but few if any have been used in attribute filtering.

Here, we propose a new method for contrast-sensitive attribute filters based on hyperconnectivity [2], [3]. We use it to cluster connected regions along the intensity range rather than spatially as in clustering-based second-generation connectivity [7], [29]. Our method is based on k -flat zones. These are defined as connected regions of maximal extent in which the total gray-level variation is no more than k . This restriction to gray-level range automatically restricts the size to which the regions can grow, as shown in Fig. 2. This does yield overlapping pseudoflat zones, and thus a cover of the image domain, but we will show that this does not prevent the definition of attribute filters. As can be seen in Fig. 3a, the new method greatly improves enhancement of internal details without distortion while suppressing stellar detail.

A different method with similar aims was proposed independently by Naegel et al. [30] and Purnama et al. [31]. Both change the filtering rule of Max-Tree-based attribute filters by setting it to preserve everything (higher level details) within any region preserved by regular attribute filters. The drawback of this approach is reduced suppression of background detail (see Fig. 3b).

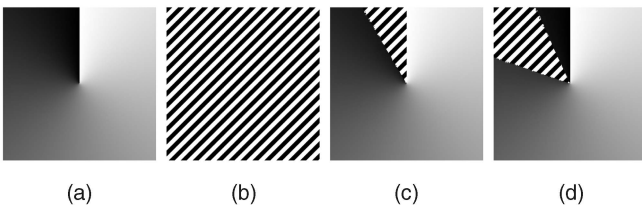


Fig. 2. The difference between k -flat zones and quasi-flat zones. (a) Image showing circular gradient. (b) Quasi-flat zone (hatched) for slope $s = 1$. (c) and (d) Two k -flat zones for $k = 16$. Note how the two k -flat zones overlap.

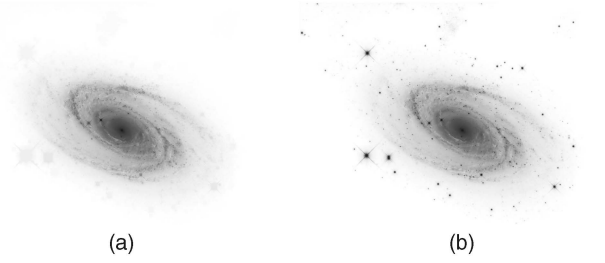


Fig. 3. Attribute filtering with k -flat zones versus branches filtering. (a) Filtered result using the k -flat version of area attribute filter used in Fig. 1d, showing improved retention of internal detail of galaxy. (b) Using the branches filtering rule which restores too much stellar detail.

The rest of the paper is organized as follows: In Section 2, we briefly present some connectivity concepts and discuss attribute filters, ways for extending them to gray scale, and computing them using the Max-Tree algorithm [27]. In Section 3, we start with a short introduction on hyperconnectivity, prove that attribute filters extend consistently to hyperconnectivity, and present the proposed method. An algorithm, together with an implementation analysis for a suitably adopted filtering rule, is given in Section 3.4. Experiments on 2D images and 3D data sets, together with a discussion on our findings, are given in Sections 4 and 5, respectively. The most important result in these sections is that in filtering historical documents, hyperconnected filters outperform their connected counterparts and methods based on anisotropic diffusion [32], [33] in terms of quality. In terms of CPU-times, the new method is two orders of magnitude faster than anisotropic diffusion. In Section 6, we summarize our conclusions.

2 CONNECTIVITY AND ATTRIBUTE FILTERS

In the following, binary images X are considered to be subsets of some universal set E , usually \mathbb{Z}^2 or \mathbb{R}^2 . Throughout, \wedge and \vee will denote logical AND and OR, respectively.

2.1 Connectivity Classes

In discrete image analysis, connectivity provides the means to group pixels into meaningful structures or objects. In mathematical morphology, a common way of addressing connectivity is through *connectivity classes* or *connections* [12], [13].

Definition 1. Let E be an arbitrary nonempty set. A *connectivity class* or *connection* \mathcal{C} on E is any family in $\mathcal{P}(E)$ that satisfies:

1. $\emptyset \in \mathcal{C}$ and for all $x \in E$, $\{x\} \in \mathcal{C}$,
2. for any $\{C_i\} \subseteq \mathcal{C}$, $\bigcap_i C_i \neq \emptyset \Rightarrow \bigcup_i C_i \in \mathcal{C}$.

Any member of \mathcal{C} is said to be *connected*. The definition means that both the empty set and singleton sets, denoted as $\{x\}$, are connected, and that any union of sets C_i in \mathcal{C} which have a nonempty intersection is also connected. The members of \mathcal{C} are called *connected sets* and correspond to subsets of E .

Given a point $x \in E$, the *connected component* or *grain* C_x of a set X can be extracted by a *connectivity opening* Γ_x at point x , which is an operator defined as

$$\Gamma_x(X) = \bigcup_j \{C_j \in \mathcal{C} \mid x \in C_j, C_j \subseteq X\}, \quad (1)$$

for every $X \subseteq E$. This yields the largest connected subset of X containing x . Connectivity openings are algebraic openings and for a family $\{\Gamma_x, x \in E\}$, there exists a one-to-one correspondence to a connectivity class on $\mathcal{P}(E)$ [2], [12], [23].

2.2 Attribute Filters

As in [15], we will call a morphological operator a filter if it is idempotent, i.e., $\Psi(\Psi(X)) = \Psi(X)$. Attribute filters [4] come in two important varieties: increasing and nonincreasing. A filter Ψ is increasing if for any two sets $X, Y \subseteq E$, $X \subseteq Y$ implies $\Psi(X) \subseteq \Psi(Y)$. In the case of connected operators, a filter $\Psi: \mathcal{P}(E) \rightarrow \mathcal{P}(E)$ interacts with connected components rather than individual pixels. In attribute filters, connected components are preserved unmodified if they meet some prespecified attribute criterion Λ or removed otherwise.

Attribute criteria for connected components $C_x \subseteq X$ are typically given in the form of

$$\Lambda(C_x) = (\text{Attr}(C_x) \geq \lambda), \quad (2)$$

with Attr some real-value attribute of C_x , and λ an attribute threshold. They are implemented by means of a trivial opening [12] which is an operator $\Gamma_\Lambda: \mathcal{C} \rightarrow \mathcal{C}$. For a connected component C_x , Γ_Λ returns C_x if $\Lambda(C_x)$ is true and \emptyset otherwise.

The attribute filter Ψ^Λ of a set X given a criterion Λ is given by

$$\Psi^\Lambda(X) = \bigcup_{x \in X} \Gamma_\Lambda(\Gamma_x(X)). \quad (3)$$

Let \mathcal{C}_X^* denote the set of all connected components of X . We can then rewrite (3) as

$$\Psi^\Lambda(X) = \bigcup_{C_j \in \mathcal{C}_X^*} \Gamma_\Lambda(C_j). \quad (4)$$

Anti-extensive attribute filters, i.e., $\Psi^\Lambda(X) \subseteq X$, are referred to as attribute openings or thinnings, depending on whether the criterion is, respectively, increasing or not. A frequently used increasing attribute is the area of connected components [34], [35]. This yields an increasing criterion if the form of (2) is used. In Fig. 1, an example of a nonincreasing criterion based on area is used:

$$\Lambda(C_x) = (\lambda_{\min} \leq A(C_x) \leq \lambda_{\max}). \quad (5)$$

Nonincreasing criteria [4], [15] often rely on scale-invariant shape descriptors such as the moments invariants. Examples are noncompactness, sparseness, elongation, sphericity, etc. [11].

2.3 Extensions to Gray Scale

Increasing connected filters extend to gray scale by threshold superposition [36]. Given a gray-scale image $f: E \rightarrow \mathbb{R}$, thresholding f in an increasing order from $h_{\min} + 1$ to h_{\max} yields a stack of nested binary sets. Each binary image at level h is given by

$$T_h(f) = \{x \in E \mid f(x) \geq h\}, \quad (6)$$

and for any two levels, $h_1 < h_2 \Rightarrow T_{h_1}(f) \subseteq T_{h_2}(f)$. Given a threshold decomposition of f , the response of the gray-scale

counterpart of a binary increasing filter Ψ^Λ on each point x of f is given by

$$\psi^\Lambda(f)(x) = \sup\{h \mid x \in \Psi^\Lambda(T_h(f))\}. \quad (7)$$

Thus, the operator ψ^Λ assigns to each x the highest threshold at which it still belongs to a connected foreground component which satisfies an attribute criterion Λ . Attribute filters are implemented efficiently on image representation structures with the aid of *filtering rules*. Depending on the rule, nonincreasing attributes can also be used to define gray-scale nonincreasing filters [4]. Filtering rules are discussed in more detail in Sections 2.4 and 3.3 and in [27], [37].

Next, we define three types of components that used for defining connected filters in gray scale. Given a gray-scale image f , a peak component P_h is a connected component of the threshold set at level h [1], [27], i.e., for some x ,

$$P_h^x(f) = \Gamma_x(T_h(f)). \quad (8)$$

Gray-scale attribute openings act on peak components, while connected filters in general act on flat zones [1]. A flat zone F_h is a connected component of the set of pixels with level strictly equal to h [1], i.e.,

$$F_h^x(f) = \Gamma_x(\{x \in E \mid f(x) = h\}). \quad (9)$$

If a flat zone F_h at level h has no neighbors of intensity greater than h , it is called a *regional maximum* M_h . All digital images have a finite number of each type of components at each gray level, and they will be indexed as P_h^i, F_h^j , and M_h^n when necessary for clarity, rather than be indicated by some point x .

2.4 The Max-Tree Algorithm

Attribute filters have been implemented efficiently with tree-based algorithms for gray-scale image representation [10], [27], [38]. An example is the Max-Tree introduced by Salembier et al. [27] for anti-extensive attribute filtering. Extensive filters are easily defined by duality using Min-Trees. The Max-Tree algorithm is a three-stage process in which the construction of the tree and the computation of node attributes is independent of filtering and image restitution. Given a gray-scale image f , the tree structure reflects the nesting order of its peak components P_h^i . The nodes $C_{h,i}^i$, addressed by their level h and index i , correspond to sets of pixels for which there exists a unique mapping to peak components:

$$C_{h,i}^i = \{x \in P_h^i \mid f(x) = h\}. \quad (10)$$

Each Max-Tree node, except for the root, points to its parent at level $h' < h$. The root node at level h_{\min} points to itself. This linking property simplifies the computation of peak component attributes since every parent inherits the auxiliary data for attribute computation of its descendents. In the case of increasing attributes such as area or volume, inheritance is a simple accumulation, while for the more complicated case of shape attributes, like the one in [37], the process relies on more sophisticated attribute handling functions described in [7], [11], [37]. Max-Tree nodes contain four important fields: *Level* and *NewLevel* for storing the

original level h and the new value after filtering, *Attribute* containing the attribute measure (in terms of auxiliary data), and *Parent* which is the parent's node address.

The tree is constructed recursively [27]. A flooding function using hierarchical first-in first-out (FIFO) queues upon receiving a pixel updates the auxiliary data buffer of the node currently being flooded. It then inspects its neighbors and places them in the appropriate queues. If a neighboring pixel is at a higher level $h' > h$, flooding proceeds recursively at h' . Once a node is fully flooded, it is finalized by determining its *Parent* and *Attribute* fields. The function returns the auxiliary data to the parent node and flooding continues at that level. The process terminates when all pixels have been processed. Implementation details are given in [7], [27], other algorithms can be found in [39], [40].

The filter function, which is called after flooding, visits each node starting from the root. For every node, the *Attribute* field is used to evaluate Λ , usually by comparison to a prespecified attribute threshold λ . Nodes failing the criterion are removed by lowering the gray level of their member pixels in accordance to the filtering rule chosen [4], [27], [31], [37]. Restitution simply assigns the new levels to the corresponding pixels.

The simplest filtering rule implements (7) directly. In the following discussion, the focus is on the *subtractive filtering rule* [37] for its simple and consistent handling of non-increasing criteria. Briefly, in the case of standard connectivity, if a peak component does not meet the criterion Λ , its flat zones are lowered in gray level to that of its highest surviving ancestor. Furthermore, it also lowers the intensity of its descendants by the same amount. We can formalize this by first introducing the characteristic function χ for a binary image X :

$$(\chi(X))(x) = \begin{cases} 1, & \text{if } x \in X, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

The subtractive rule then becomes

$$\psi^\Lambda(f)(x) = \int_{h=0}^{\infty} (\chi(\Psi^\Lambda(T_h(f))))(x) dh. \quad (12)$$

This means that the supremum in (7) is replaced by integrating the characteristic function of the filtering results on each threshold set. In practice, integration is replaced by summation in discrete images. In the following sections, we will show how this approach can be extended to hyperconnectivity.

3 HYPERCONNECTIVITY

We will now introduce hyperconnectivities, which will lead to a consistent interpretation of k -flat zones within the context of attribute filters.

3.1 Hyperconnectivity Classes

Hyperconnectivity [2], [3] extends the notion of connectivity by relaxing the second condition of Definition 1. Instead of using a strict nonempty intersection of sets for their union to be connected, the definition of *hyperconnectivity classes* involves a degree of overlap specified by an *overlap criterion*.

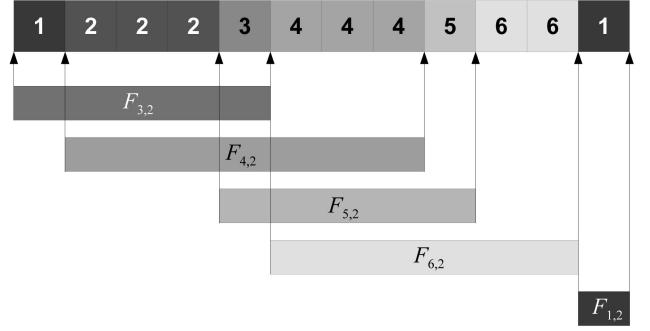


Fig. 4. A signal and its 2-flat zones: Each shaded box and associated arrows indicate a 2-flat zone. These zones are not disjoint.

Definition 2. An overlap criterion in $\mathcal{P}(E)$ is a mapping $\perp : \mathcal{P}(\mathcal{P}(E)) \rightarrow \{0, 1\}$ such that \perp is decreasing, i.e., for any $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}(E)$,

$$\mathcal{A} \subseteq \mathcal{B} \Rightarrow \perp(\mathcal{B}) \leq \perp(\mathcal{A}). \quad (13)$$

Any $\mathcal{A} \subseteq \mathcal{P}(E)$ for which $\perp(\mathcal{A}) = 1$ is said to be overlapping; otherwise, \mathcal{A} is nonoverlapping. The condition (13) ensures that a nonoverlapping family cannot possibly become overlapping by adding more sets. A hyperconnectivity class can be defined as follows:

Definition 3. A hyperconnectivity class $\mathcal{H} \subseteq \mathcal{P}(E)$ with an overlap criterion \perp is a family of sets with the following properties:

1. $\emptyset \in \mathcal{H}$ and for all $x \in E$, $\{x\} \in \mathcal{H}$,
2. for any $\{A_j\} \subseteq \mathcal{H}$ for which $\perp(\{A_j\}) = 1 \Rightarrow \bigcup A_j \in \mathcal{H}$.

Members of a hyperconnectivity class are called hyperconnected. It can be seen that all connectivity classes are special cases of hyperconnectivity [3] in which the overlap criterion is given by

$$\perp(\{A_j\}) = \begin{cases} 1, & \text{if } \bigcap A_j \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

For a more practical example of an overlap criterion in this context, consider the notion of k -flat zones, defined informally in Section 1. This example is directly relevant because we are going to design our filters around them. A more formal definition is the following:

Definition 4. A k -flat zone $F_{h,k}$ at level h and depth k is the set of all pathwise connected pixels, marked from $x \in E$, with intensities from h down to $h - k$:

$$F_{h,k}(x) = \Gamma_x(\{p \in E \mid h - k \leq f(p) \leq h\}). \quad (15)$$

A 1D example is shown in Fig. 4. It is obvious from Definition 4 and Fig. 4 that for $k > 0$, the k -flat zones found at all levels $h \in [h_{\min} + k, h_{\max}]$ show overlap, and thus do not form a partition of E . Consequently, they do not represent a connectivity on E in any way. We can derive a hyperconnectivity class from k -flat zones.

Proposition 1. Let $f : E \rightarrow \mathbb{ID}$, with \mathbb{ID} the totally ordered universe of gray values be a gray-scale image, with associated k -flat zones $F_{h,k}$ with $k \in \mathbb{ID}$. This induces a hyperconnectivity class given by

$$\mathcal{H}_f^k = \{\emptyset\} \cup \{A \in \mathcal{C} \mid \|f(p) - f(q)\| \leq k \ \forall p, q \in A\} \quad (16)$$

that is associated with the overlap criterion from:

$$\perp_f^k(\{H_j\}) = \begin{cases} 1, & \text{if } \bigcap H_j \neq \emptyset \wedge \max_{p, q \in \bigcup H_j} \|f(p) - f(q)\| \leq k, \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

This proposition states that any connected subset of a k -flat zone of f is a member of a hyperconnectivity class \mathcal{H}_f^k . Furthermore, two such sets overlap iff their union is connected according to some connectivity class \mathcal{C} , and the gray-level range within image f over the union does not exceed k .

Proof. We observe that $\emptyset \in \mathcal{H}_f^k$ by definition. Furthermore, $\{x\} \in \mathcal{H}_f^k$ for all $x \in E$ because they are members of the “base” connectivity class \mathcal{C} and the gray-level range within a singleton is zero. This proves that \mathcal{H}_f^k meets the first condition of Definition 3. Moreover, from (15) and (16), it is obvious that any $F_{h,k} \in \mathcal{H}_f^k$.

We must now prove that \perp_f^k is decreasing. Let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{H}_f^k$, with $\mathcal{A} \subseteq \mathcal{B}$. Let $\perp_f^k(\mathcal{B}) = 1$. This means that for the intersection of all elements of \mathcal{B} , we have $\bigcap \mathcal{B} \neq \emptyset$ and because $\mathcal{A} \subseteq \mathcal{B} \Rightarrow \bigcap \mathcal{A} \neq \emptyset$. Furthermore, $\perp_f^k(\mathcal{B}) = 1$ means that the gray-level range in f within the union of all elements of \mathcal{B} denoted as $\bigcup \mathcal{B}$ is smaller than k . However, $\bigcup \mathcal{A} \subseteq \bigcup \mathcal{B}$, and therefore the gray-level range must be smaller than or equal to k too. Thus,

$$\mathcal{A} \subseteq \mathcal{B} \Rightarrow (\perp_f^k(\mathcal{B}) \Rightarrow \perp_f^k(\mathcal{A})), \quad (18)$$

proving that it is decreasing.

We must now prove that if $\perp_f^k(\{H_j\}) = 1$ and $\{H_j\} \subseteq \mathcal{H}_f^k \subseteq \mathcal{C}$, then $\bigcup H_j \in \mathcal{H}_f^k$. If $\perp_f^k(\{H_j\}) = 1$, the intersection $\bigcap H_j$ is not empty, and therefore $\bigcup H_j \in \mathcal{C}$. Furthermore, the gray-level range within the union $\bigcup H_j \leq k$ if $\perp_f^k(\{H_j\}) = 1$, proving that $\perp_f^k(\{H_j\}) = 1$ which in turn implies that $\bigcup H_j \in \mathcal{H}_f^k$, according to (16). \square

If k is set to 0, the k -flat zones become the ordinary flat zones of f , and \mathcal{H}_f^k becomes a connectivity class.

Note that the k -flat zones are the maximal elements of \mathcal{H}_f^k , i.e., there are no elements $H' \in \mathcal{H}_f^k$ such that $F_{h,k} \subset H'$ for any $F_{h,k}$. This is equivalent to stating that $\perp_f^k(\{F_{h,k}^i, F_{h',k}^j\}) = 1$ implies that $i = j$ and $h = h'$, i.e., $F_{h,k}^i = F_{h',k}^j$. In the following, we will show that such maximal hyperconnected sets lead to a natural extension of attribute filters to hyperconnectivity.

3.2 Hyperconnected Components and Attribute Filters

Let us return to the binary case in which images are subsets of E . Given a binary image X and associated hyperconnectivity class \mathcal{H} , let

$$\mathcal{H}_X = \{A \in \mathcal{H} \mid A \subseteq X\} \quad (19)$$

be the family of all hyperconnected subsets of X . We can then extract the maximal elements of \mathcal{H}_X by removing any hyperconnected set which has a hyperconnected superset within X . Members of \mathcal{H}_X^* , given by

$$\mathcal{H}_X^* = \{A \in \mathcal{H}_X \mid \nexists B \in \mathcal{H}_X : A \subset B\}, \quad (20)$$

are hyperconnected sets of maximal extent, i.e., *hyperconnected components*. Two important observations are: 1) for $H_j \in \mathcal{H}_X^*$:

$$\bigcup_{H_j \in \mathcal{H}_X^*} H_j = X, \quad (21)$$

and 2):

$$\perp_f^k(\{H_i, H_j\}) = 1 \Rightarrow H_i = H_j, \quad (22)$$

for all $H_i, H_j \in \mathcal{H}_X^*$. Hyperconnected attribute filters $\Psi_{\mathcal{H}}^{\Lambda}$ can be defined as in (4) by replacing \mathcal{C}_X^* with \mathcal{H}_X^* .

Proposition 2. Let \mathcal{H} be a hyperconnectivity class on E and \mathcal{H}_X^* be the family of hyperconnected components of an image $X \subseteq E$. For any increasing criterion Λ , $\Psi_{\mathcal{H}}^{\Lambda}$, given by

$$\Psi_{\mathcal{H}}^{\Lambda}(X) = \bigcup_{H_j \in \mathcal{H}_X^*} \Gamma_{\Lambda}(H_j), \quad (23)$$

is an algebraic opening.

Proof. Algebraic openings have three distinct properties: They are anti-extensive, increasing, and idempotent. $\Psi_{\mathcal{H}}^{\Lambda}$ is anti-extensive and so is Γ_{Λ} , i.e., $\Psi_{\mathcal{H}}^{\Lambda}(X) \subseteq X$ because

$$\Psi_{\mathcal{H}}^{\Lambda}(X) = \bigcup_{H_j \in \mathcal{H}_X^*} \Gamma_{\Lambda}(H_j) \subseteq \bigcup_{H_j \in \mathcal{H}_X^*} H_j = X. \quad (24)$$

Let $X, Y \subseteq E$ be two images such that $X \subseteq Y$. To prove that $\Psi_{\mathcal{H}}^{\Lambda}$ is increasing, consider any hyperconnected component $H_j \in \mathcal{H}_X^*$. Obviously, $H_j \subseteq Y$, and thus $H_j \in \mathcal{H}_Y$. However, $H_j \in \mathcal{H}_Y^*$ is not necessarily true, i.e., H_j needs not be a hyperconnected component of Y . Given that Λ is increasing, if $H_j \notin \mathcal{H}_Y^*$ and $\Lambda(H_j) = \text{true}$, there exists some hyperconnected component of Y , say H_i , which is a superset of H_j and $\Lambda(H_i) = \text{true}$. Therefore, for any $H_j \in \mathcal{H}_X^*$,

$$H_j \subseteq \Psi_{\mathcal{H}}^{\Lambda}(X) \Rightarrow H_j \subseteq \Psi_{\mathcal{H}}^{\Lambda}(Y), \quad (25)$$

because $\Lambda(H_j) \Rightarrow \Lambda(H_i)$. If $H_j \in \mathcal{H}_Y^*$, then $H_i = H_j$ and (25) is true, as required. In both cases, $\Psi_{\mathcal{H}}^{\Lambda}$ is increasing.

To prove that $\Psi_{\mathcal{H}}^{\Lambda}$ is idempotent, first consider that $\Psi_{\mathcal{H}}^{\Lambda}(X)$ is the union of those hyperconnected components of X which satisfy Λ , as seen in (23). Due to (22), none of them overlaps, and therefore all surviving components are also the hyperconnected components of $\Psi_{\mathcal{H}}^{\Lambda}(X)$: No new components can be created. Thus,

$$\Gamma_{\Lambda}(H_j) = H_j \quad \forall H_j \in \mathcal{H}_{\Psi_{\mathcal{H}}^{\Lambda}(X)}^*, \quad (26)$$

i.e., all hyperconnected components of $\Psi_{\mathcal{H}}^{\Lambda}(X)$ pass Λ . Therefore,

$$\begin{aligned} \Psi_{\mathcal{H}}^{\Lambda}(\Psi_{\mathcal{H}}^{\Lambda}(X)) &= \bigcup_{H_j \in \mathcal{H}_{\Psi_{\mathcal{H}}^{\Lambda}(X)}^*} \Gamma_{\Lambda}(H_j) \\ &= \bigcup_{H_j \in \mathcal{H}_{\Psi_{\mathcal{H}}^{\Lambda}(X)}^*} H_j = \Psi_{\mathcal{H}}^{\Lambda}(X), \end{aligned} \quad (27)$$

proving idempotence. \square

If Λ is nonincreasing, $\Psi_{\mathcal{H}}^{\Lambda}$ is idempotent and anti-extensive according to the same reasoning, and $\Psi_{\mathcal{H}}^{\Lambda}$ becomes

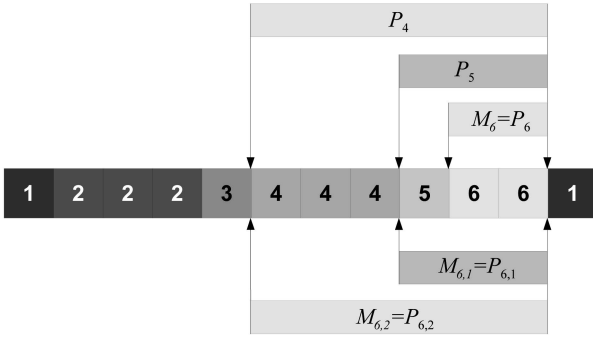


Fig. 5. The equivalence of k -regional maxima $M_{h,k}$, with regular peak components P_{h-k} : a signal and its k -regional maxima ($k = 1, 2$) at $h = 6$, and its regular peak components at $h = 4, 5, 6$. In all cases, $M_{h,k} = P_{h-k}$.

an attribute thinning. Thus, hyperconnected attribute filters preserve the main properties of connected attribute filters given in [4]. Computing them, however, has a major bottleneck in the extraction of hyperconnected components. In [3], Braga-Neto and Goutsias defined a hyperconnectivity opening η in analogy to regular connectivity openings. Given a point $x \in E$, $\eta(X)$ extracts the union of all hyperconnected sets of X containing x . The union, however, may not necessarily be a hyperconnected set itself due to the overlap criterion. Defining the axiomatics of operators that can extract hyperconnected components remains an open problem and it is not dealt with in this paper. For the purposes of our work, we employ k -flat zones, which can be computed readily.

3.3 Attribute Filters Based on k -Flat Zones

As was discussed in Section 2.4, attribute filters for gray-scale images work on peak components and can readily be computed using Max-Tree [27]. The attribute of each peak component is evaluated using criterion Λ , and its gray level is adjusted according to the filtering rule chosen [4], [27], [37]. If regular peak components are replaced with k -peak components, contrast information can be incorporated in the filter's decision making by controlling the parameter k . That allows the preservation of fine internal structure which fails attribute criterion Λ within objects of interest, while suppressing all low contrast structures, like noise in the unwanted background, even if they satisfy Λ .

Let us now consider the k -flat zone equivalent of regional maxima. A regional maximum M_h at level h is a 0-flat zone which has neighbors of strictly lower gray level. Equivalently, a k -regional maximum $M_{h,k}$ is a k -flat zone at level h which has neighbors of gray level strictly smaller than $h - k$. Obviously,

$$M_h = M_{h,0} \subseteq M_{h,k} \quad \forall k > 0. \quad (28)$$

These same regional maxima correspond to k -peak components $P_{h,k}$ at their respective gray levels. This means that these k -peak components correspond exactly to regular peak components at level $h - k$, i.e.,

$$M_{h,k} = P_{h,k} = P_{h-k}. \quad (29)$$

An example is shown in Fig. 5. If we extend this equivalence of k -peak components to 0-peak components from only the

regional maxima to all peak components, we can define a Max-Tree based on k -peak components. If $k = 0$, we end up with the regular Max-Tree (as it should be), but, as k is increased, we will change the topology of the tree. In particular, we will cluster multiple peak components P_h^i into their supersets $P_{h,k}^j = P_{h-k}^j$, with i and j some indices. Note, however, that these clusters overlap. In principle, such clustering can be performed by the dual-input Max-Tree algorithm intended for second-generation connectivities [7], using a mask m defined as

$$m(x) = f(x) + h \quad \forall x \in E. \quad (30)$$

Though this approach would work, it is not practical for two reasons. First, it is wasteful in terms of memory. Second, this approach implicitly assumes that the base connectivity class is regular 4 or 8 connectivity. The theory in Section 3.1 makes no such assumption and allows *any* connectivity class to be used. Therefore, we will build a regular Max-Tree, either using the original algorithm [27], if 4 or 8 connectivity is to be used, or the dual-input algorithm [7] for spatial clustering or partitioning second-generation connectivities. We will implement k -flat filtering as a new filtering rule for this Max-Tree. This approach also allows changing k after the Max-Tree has been built, allowing interactive changing of the value of k , combined with visualization of the results in the case of 3D data, similar to [11].

Consider now a threshold decomposition of a gray-scale image f . The component $P_{h,k}$ is preserved if $\text{Attr}(P_{h-k}) \geq \lambda$ and this decision propagates to all regular peak components from P_{h-k} up to P_h , independent of whether each one of them separately satisfies Λ or not. Obviously, k -peak components are defined from k levels above the background until the maximum level. Regular peak components $P_{h'}$ which are not members of some k -peak component, i.e., $P_{h'+k} = \emptyset$, are rejected because of low contrast.

Therefore, given any criterion Λ for regular Max-Tree filtering, we can derive a modified criterion Λ_k for k -flat Max-Tree filtering as

$$\Lambda_k(P_h) = \begin{cases} \Lambda(P_h), & \text{if } \exists x \in P_h : f(x) \geq h + k, \\ 0, & \text{otherwise.} \end{cases} \quad (31)$$

For any peak component, this first checks if the contrast within it is sufficient, and only then computes $\Lambda(P_h)$. However, modifying the criterion is not sufficient for our purpose. Let P_h be preserved because it meets Λ_k , but its descendants do not. If we fail to restore these descendants up to a level k above the current P_h , a second application of the filter would remove P_h because it would automatically fail Λ_k and this would violate idempotence.

Therefore, we must modify the classical subtractive Max-Tree filtering rule so that we propagate preserve decisions based on Λ_k upward within a range k above the preserved ancestors. To do this, we add a field k' to every Max-Tree node to indicate the current propagation range. For any node preserved because Λ_k is met, its k' is set to k . Suppose we are currently inspecting node $P_{h'}$, with parent P_h (with $h < h'$ obviously), and let k'_h be the propagation range set for P_h and $\Delta h = h' - h$. If $\Lambda_k(P_{h'})$ is false, but $\Delta h \leq k'_h$, and thus it falls within the propagation range, it is also preserved, and its propagation range $k'_{h'}$ is set to $k'_h - \Delta h$. In the subtractive

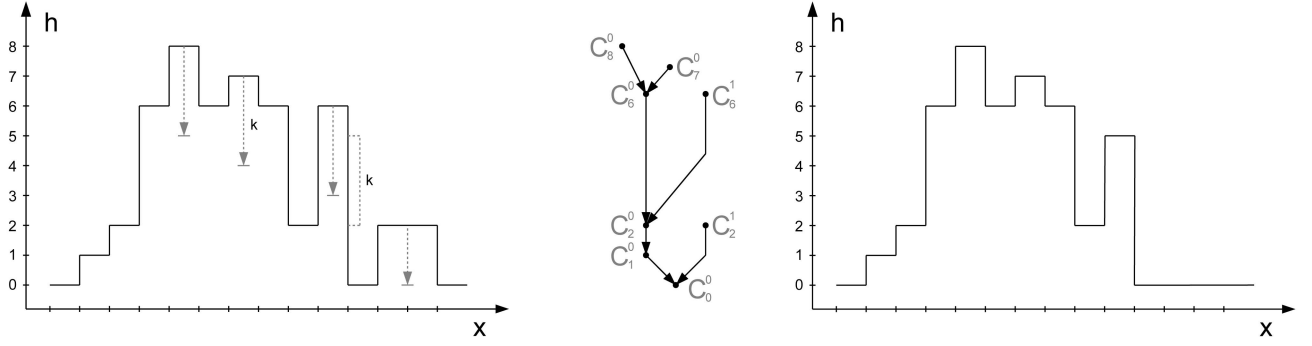


Fig. 6. The k -subtractive filtering rule for a 1D signal. (from left) The original 1D signal (solid line) with the four regional maxima; the Max-Tree of the original signal; the filtered output using the k -subtractive filtering rule.

rule, any preserved node receives a gray level Δh above the new level assigned to its parent. We must therefore start at the root and proceed upward in this process.

However, if $\Lambda_k(P_{h'})$ is false and $\Delta h > k_{h'}$, the node is removed and its propagation range $k_{h'}$ is set to zero. Normally in the subtractive rule, its new gray level would be set to its parent's new gray level. In the k -subtractive case, we do the same, but add its parent's propagation range k_h to this value. This ensures that for any node P_h preserved because $\Lambda_k(P_h)$ is true, at least a range of nodes up to a level $h + k$ are preserved as well and, in subsequent filtering at the same value of k , it will again be preserved (albeit at its new gray level).

We summarize this set of rules to what we call the k -subtractive filtering rule, which is defined as follows:

Definition 5. A gray-scale attribute filter ψ_k^Λ configured with the k -subtractive filtering rule is given by

$$(\psi_k^\Lambda(f))(x) = h_{\min} + \sum_{h=h_{\min}}^{h_{\max}} \sum_{j \in J_h^f} \chi(\Gamma_\Lambda^k(P_h^j)), \quad (32)$$

in which J_h^f is an index set for peak components at level h of an image f , and Γ_Λ^k is defined as

$$\Gamma_\Lambda^k(P_h^j) = \begin{cases} P_h^j, & \text{if } \Lambda_k(P_h^j) \vee \exists P_{h'}^l \supset P_h^j : \\ & (h' \geq h - k \wedge \Lambda_k(P_{h'}^l)), \\ \emptyset, & \text{otherwise.} \end{cases} \quad (33)$$

Implementation details of the k -subtractive filtering rule and a pseudocode are given in Section 3.4.

A 1D example is given in Fig. 6. Assume that we use a nonincreasing criterion and $k = 3$. In the original signal, there exist four regional maxima for which we assume that the first three from the left, i.e., P_8^0 , P_7^0 , and P_6^1 , fail Λ , while $\text{Attr}(P_2^1) > \lambda$. Also, all of the other peak components, except for the root, satisfy Λ . An attribute filter relying on standard connectivity and configured with the subtractive rule would remove the first three regional maxima by lowering their intensities to those of their respective parents and leave the rest of the signal unaffected. In the case in which the same filter is configured with the k -subtractive filtering rule instead, the results vary significantly. Starting with the background, we see that since it fails Λ and there is no other node below it, it is rejected, with its propagation range set to 0. The left lobe of the signal has two k -peak components before it

gets split in two. Both $P_{4,3}^0$ and $P_{5,3}^0$ are preserved because they have sufficient contrast and P_1^0 and P_2^0 , respectively, satisfy Λ . P_2^0 propagates a preserve status for three levels up and, though this does not affect much in the left group of descendants, it preserves part of the right group (gray-dotted arrow) in which all descending peak components fail Λ . That is, the previously rejected regional maximum P_6^1 is only lowered by 1. On the left group again, we keep on finding peak components that satisfy Λ until level $h = 5$. From $h = 6$ and up, the contrast range does not permit for any more and the decision on the remaining regular peak components is based on the youngest surviving k -peak component, i.e., $P_{8,3}^0 = P_5^0$. Since they are all within the contrast range of $P_{8,3}^0$ (black-dotted arrow), they are preserved as they are. Coming back to the background component, we see that the contrast range of the right lobe is below k ; thus, there cannot be any k -peak components. The decision of the regional maximum P_2^1 is thus left on the background component which was rejected and with its propagation range set to 0, i.e., P_2^1 , though it satisfies Λ , is rejected.

3.4 The k -Subtractive Filtering Algorithm

The k -subtractive filtering rule defined in Section 3.3 is implemented as a separate function that takes as input a Max-Tree structure and the two parameters λ and k . It requires a single pass through the tree.

In this version of the Max-Tree, two extra fields per node are required: *PeakLevel* and *kprime*, which store the gray level of the descendent with the highest gray level and the propagation range, respectively. The first is initialized to each node's original level and the second to 0 while constructing the tree. Prior to filtering, the *SetPeakLevels()* function is called. This routine traverses the tree from the leaves to the root, and at each node, sets the *PeakLevel* field of the parent to its own *PeakLevel* if that is higher than the parent's *PeakLevel*. This ensures that the *PeakLevel* field of each node C_h^i is set to the maximum within the corresponding peak component P_h^i .

The *PeakLevel* is used to determine whether each node examined associates to a k -peak component or not. That is, if the difference between the node's *PeakLevel* and its parent's level is greater than or equal to k , then there is sufficient contrast range and the corresponding P_h defines a $P_{h,k}$. The *kprime* member specifies the propagation range, i.e., which nested peak components are to be preserved

Algorithm 1 The k -Subtractive filtering rule.

```

procedure  $kSubtractive$  (var  $t$ :MaxTree ,  $\lambda$ :AttributeType,  $k$ :GreyLevel)
  /* input: Max-Tree  $t$  of an image  $f$ , attribute threshold  $\lambda$  and  $k$ -flat value */

  var  $root$ ,  $node$ ,  $parnode$  ; /* type MaxNode, pointers to root, current node and its parent */
  var  $l$ ,  $difflevel$  ;      /* type GreyLevel, level and level difference */

  /* process root */
   $root = t.root$ 
   $l = root.Level$  /* sets  $l = h_{min}$  */
  if  $root.Attribute > \lambda$  then
     $root.kprime = k$  /* preserve and set maximal  $k$ -restoration level */
     $root.NewLevel = root.Level$ 
  else
     $root.kprime = 0$ 
     $root.NewLevel = 0$ 
  endif
  for all levels  $l$  starting at  $h_{min} + 1$  to  $h_{max}$  do
    for all nodes  $node$  at each level  $l$  do
       $parnode = node.parent$ 
       $difflevel = node.Level - parnode \rightarrow Level$ 
      if  $node.PeakLevel - parnode.Level > k$  AND  $node.Attribute > \lambda$  then
        /* preserve and set maximal  $k$ -restoration level */
         $node.NewLevel = parnode.NewLevel + difflevel$ 
         $node.kprime = k$ 
      else
        if  $difflevel > parnode.kprime$  then
           $node.NewLevel = parnode.NewLevel + parnode.kprime$ 
           $node.kprime = 0$ 
          /*  $k$ -restoration completed */
        else
           $node.NewLevel = parnode.NewLevel + difflevel$ 
           $node.kprime = parnode.kprime - difflevel$ 
          /* remainder to be  $k$ -restored */
        endif
      endif
    endfor
  endfor
  for all pixels  $p$  restitute the filtered image

```

Section to change for k -absorption ruleFig. 7. Algorithm 1: The k -subtractive filtering rule.

independent of their attribute measure if there exists a k -peak component among their ancestors.

The function starts by reading all nodes from the root upward. The root node is handled separately since it does not carry any restore decisions from previous nodes. If it defines a k -peak component that satisfies Λ , it sets $kprime$ to the maximum range, i.e., k , otherwise to 0, as is its $NewLevel$ field. After processing the root, the rest of the nodes are scanned from the root upward. If they meet the criterion Λ , their $NewLevel$ and $kprime$ fields are set as in the case of the root. If criterion Λ is not met or if the difference between their $PeakLevel$ and their parent's original level is smaller than k , there are two situations. If the difference between its gray level and its parent's is greater than the propagation range $kprime$ of its parent (which may vary from 0 to k), then it must be lowered to a new level, which is that of its parent plus the parent's propagation range. Since it is a rejected component which is out of range, its $kprime$ is set to 0, i.e., it does not carry any restore decisions from its ancestors because it is out of their range and also has nothing to propagate itself. If,

however, it is within the propagation range of some ancestor, it is preserved and updates its level to that of its parent plus the gray levels difference with it, and propagates the remaining range further up. The process terminates when all nodes have been visited. Image restitution is the same as in [27]. Variants for the other rules can be made in a similar way. The pseudocode for the k -subtractive filtering function is given in Algorithm 1 (see Fig. 7).

3.5 The k -Absorption Rule

In some applications, in particular, the historical documents studied in Section 4.1, it is useful to propagate *reject* decisions to the descendents. This scheme is very useful when we wish to absorb low-contrast background features which touch high-contrast objects into an otherwise smooth background. Only a small portion of the code indicated in Algorithm 1 needs to be changed. In this scheme, the value of $kprime$ may be positive as before, to indicate propagation of the preserve decision, but it is negative when we propagate a *reject* decision. If we find a node which is more

than k lower than its *PeakLevel* and which must be rejected based on its attribute, we set its $kprime$ to $-k$. All nodes up to k above it will be set to its *NewLevel*. However, assuming that no further children are rejected, the first node to meet the attribute criterion will be restored to its original contrast with respect to the absorbing node.

The algorithm is given in Algorithm 2. We first check whether we are more than k from a peak. If so, we check whether the attribute threshold is met. A third if-statement determines whether or not absorption is taking place. All criteria are met and preserved as before. However, if the first two are met, but the node must be absorbed, we set the current $kprime$ to that of the parent and add the difference in gray levels. If the result is positive, the node is restored according to the scheme above. If a node fails the attribute criterion, it is rejected and $kprime$ is set to $-k$. Any node with gray level h within k of its *PeakLevel* is treated as before, though we must check that negative values of $kprime$ are treated as zeros in this case.

Algorithm 2. Changes in Alg. 1 required for the k -absorption filtering rule

```

if node.PeakLevel - parnode.Level >  $k$  then
  if node.Attribute >  $\lambda$  then
    if parnode.kprime  $\geq 0$  then
      /* preserve and set maximal  $k$ -restoration level */
      node.NewLevel = parnode.NewLevel + difflevel
      node.kprime =  $k$ 
    else
      /* absorb node */
      node.kprime = parnode.kprime + difflevel
      node.NewLevel = parnode.NewLevel
      if node.kprime > 0 then /* partial absorption */
        node.NewLevel += node.kprime +  $k$ 
        node.kprime =  $k$ 
      endif
    endif
  else /* initiate absorption */
    node.NewLevel = parnode.NewLevel
    node.kprime =  $-k$ 
  endif
endif
if difflevel > parnode.kprime then
  node.NewLevel = parnode.NewLevel +
    max(parnode.kprime, 0)
  node.kprime = 0
  /*  $k$ -restoration completed */
else
  node.NewLevel = parnode.NewLevel + difflevel
  node.kprime = parnode.kprime - difflevel
  /* remainder to be  $k$ -restored */
endif
endif

```

4 EXPERIMENTS

The properties of the new filters are demonstrated in a set of experiments that follow. In each of the first four cases, the filter is computed for $k = 0$ and $k > 0$, using the same λ , so

that comparisons can be made. Since filtering is a separate stage of the Max-Tree algorithm, k , just like λ , can be adjusted interactively and in real time in 3D. The objective in all cases is to enhance the images or volumes by reducing the background and suppressing noise and other low-contrast structures superimposed on the background, while maintaining the full integrity of the targeted objects without losing internal details.

In line with the theory developed so far, k is held fixed for all nodes in Sections 4.1, 4.2, and 4.3. In Section 4.4, we extend this to variable k within the image as a function of gray level.

4.1 Historical Documents

In this experiment, the aim is to flatten the background of a document as much as possible without removing any details within the letters. The documents involved are pages from *De Agro Frisae* (1605) by Ubbo Emmius (1547-1625). In the process of digitizing this book, a problem occurred such that the letters on the reverse side are often visible as low-contrast details in the background and hamper character recognition, as evident in Fig. 8a.

Filtering with a rolling ball filter (Fig. 8b), anisotropic diffusion (Fig. 8c) [32] and an area attribute filter which retains dark structures with an area between 10 and 8,000 pixels (Fig. 8d) are able to suppress the background significantly. However, detail in the same size or shape category as letters cannot readily be removed without removing the letters themselves using such techniques. The unique combination of size and contrast information used by k -flat filters efficiently removes most of the remaining background detail, as shown in Fig. 8e. However, background details that touch the foreground letters are not removed properly. The best result is obtained with the k -absorption rule, even at a more conservative setting of $k = 30$, as shown in Fig. 8f, and the difference images in Figs. 8g, 8h, 8i, and 8j.

Setting k in this instance is fairly straightforward and was set to about half the contrast of letters (about 128 in this case). CPU-time was 0.68 s on a $2,056 \times 3,088$ gray-scale image (on a Core 2 Duo 8400 at 3.0 GHz with 2 GB of RAM).

To put these visual results in a more quantitative framework, we exported the first page of a preliminary version of this paper as a $2,478 \times 3,510$ gray-scale bitmap f_1 , with floating-point values ranging from 0 to 1. We simulated the background image f_e by exporting page 3 in a similar way, rotating it by 2 degrees, and blurring it using a Gaussian of $\sigma = 2.5$. The corrupted image f_c is computed as

$$f_c = (c_1 f_1 + (1 - c_1))(c_e f_e + (1 - c_e)) + \sigma \nu, \quad (34)$$

with c_1 the contrast factor for the foreground, $c_e = 0.25$ the contrast of the reverse-side lettering, and ν Gaussian noise with unit variance. A cropped part of such an image for $c_1 = 0.5$ and $\sigma^2 = 0.02$ can be seen in Fig. 9a. For morphological processing, the images were rescaled to 0-255, and saved as portable gray maps (pgm-files). Anisotropic diffusion was carried out on the floating-point images.

We filtered the images with five methods used in Fig. 8 and computed the Universal Quality Index (UQI) [41] and Structural Similarity Index (SSIM) [42]. The results for two

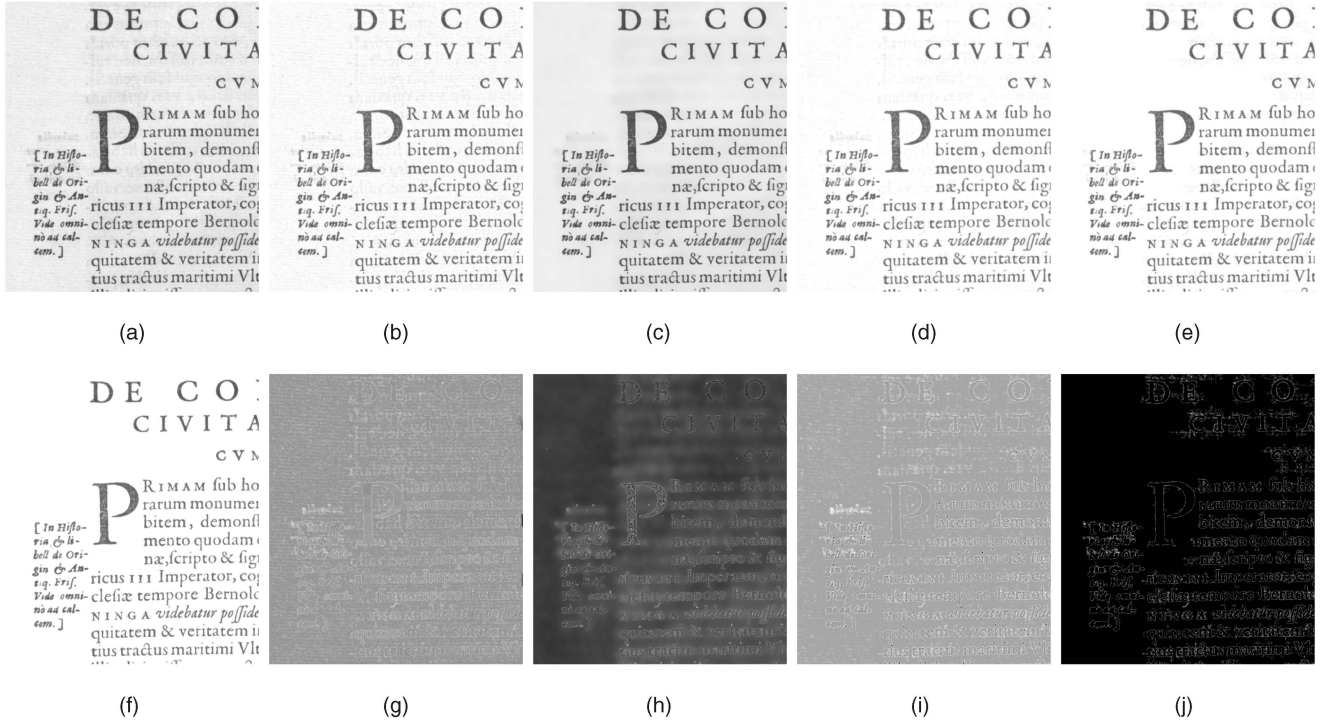


Fig. 8. Processing an historical document: (a) Original image showing much detail in the background; (b) background removed by rolling ball filter with radius 15; (c) result of anisotropic diffusion; (d) area attribute filtered with $10 \leq \text{area} \leq 8,000$ ($k = 0$), small detail in the background is still retained; (e) same filter with $k = 60$ flattens the background better; (f) k -absorption rule at $k = 30$ produces the best results, allowing suppression of details which touch the desired structures; (g), (h), (i), and (j) differences between (f) and (b), (c), (d), and (e), respectively. In the latter four, lighter areas indicate background detail suppressed in (e) but not in the other filter; darker areas indicate details within letters retained in (e) but rejected in the alternative. Original image courtesy of A. Meijster, Department of Artificial Intelligence, and the University Library, University of Groningen.

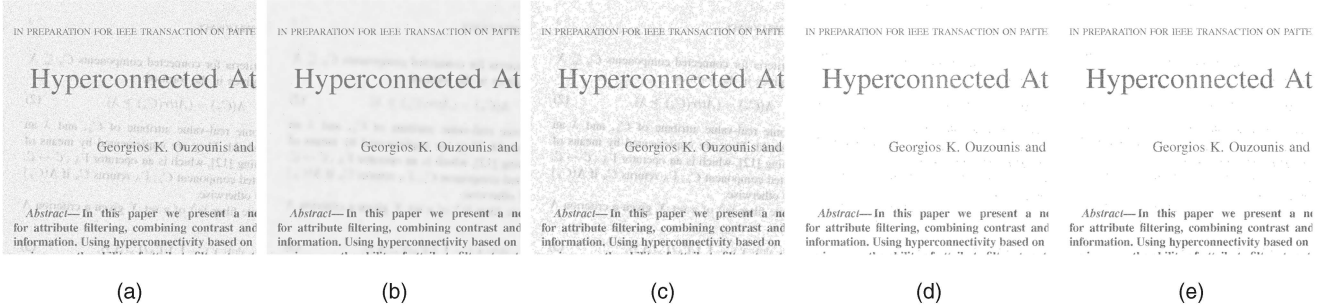


Fig. 9. Simulated document: (a) (part of) corrupted image f_c ; (b) filtered with anisotropic diffusion; (c) area attribute filtered with $5 \leq \text{area} \leq 4,000$ ($k = 80$); (d) same filter with k -absorption rule and $k = 60$ flattens the background better; (e) postprocessing with area opening improves noise suppression.

settings of c_1 and four different settings of σ are shown in Table 1. We first studied a fairly high S/N case with $c_1 = 0.85$. We obtained optimal results using $5 \leq \text{area} \leq 4,000$ for all values of k tested, for the area-based methods, and a rolling ball radius of 15 pixels. In anisotropic diffusion, we obtained the best results using edge strength threshold parameter $\lambda = 0.025$, and between 25 and 100 iterations, depending on the noise. This required between 40 and 170 s CPU time on a Core 2 Duo E8400 at 3.0 GHz with 2 GB memory, using only a single core. By comparison, the k -flat filters (all types) required between 0.37 and 1.38 s CPU time on the same machine, mainly dependent on the level of noise (which increases the size of the Max-Tree).

As can be seen in Table 1, the k -absorption filter performs best in terms of both UQI and SSIM. In terms of SSIM, the Perona and Malik method comes second because

of good noise removal. However, it performs poorly in terms of UQI, which penalizes it for failing to remove the blurred background letters fully (see Fig. 9b). The rolling ball filters are not meant to remove noise, so it performs poorly when noise is added. It also fails to remove the background letters. We therefore did not investigate this filter any further. The regular area opening ($k = 0$) performs more robustly, but fails to remove background letters. It also suffers when noise is added. The same holds for the k -subtractive method, using $k = 80$, which suffers even more because, whenever a noise feature touches a letter, even at a modest contrast level, all of its descendants up to a level k will be preserved (see Fig. 9c).

At the lower contrast setting $c_1 = 0.5$, the situation is similar, though it was harder to find optimal settings for the anisotropic diffusion method. Apart from the zero noise case, we found two different optimal settings depending on

TABLE 1
Background Suppression Results for Simulated Document

Contrast	Method	$\sigma = 0.00$		$\sigma = 0.1$		$\sigma = 0.1414$		$\sigma = 0.2236$	
		UQI	SSIM	UQI	SSIM	UQI	SSIM	UQI	SSIM
85%	no filter	0.7744	0.9418	0.2690	0.4188	0.2613	0.3388	0.2408	0.2588
	<i>k</i> -flat methods								
	$k = 0$	0.8028	0.9479	0.8038	0.9479	0.2668	0.6020	0.2521	0.4536
	$k = 80$	0.8585	0.9626	0.6311	0.7831	0.3105	0.4837	0.2449	0.2875
	<i>k</i> -absorption methods								
	$k = 80$	0.9982	0.9980	0.9948	0.9937	0.9902	0.9896	0.9606	0.9649
	Other methods								
	Anisotropic diffusion	0.7044	0.9835	0.2818	0.9861	0.2769	0.9819	0.2629	0.9037
	Rolling ball	0.7770	0.9453	0.2696	0.4314	0.2621	0.3470	0.2417	0.2630
	no filter	0.7733	0.9394	0.2526	0.3918	0.2327	0.3006	0.1887	0.2028
50%	<i>k</i> -flat methods								
	$k = 0$	0.8017	0.9461	0.2614	0.6911	0.2475	0.5746	0.2138	0.4074
	$k = 80$	0.8576	0.9609	0.6150	0.7609	0.2861	0.4510	0.1969	0.2340
	<i>k</i> -absorption methods								
	$k = 60$	0.9980	0.9981	0.9853	0.9841	0.9619	0.9634	0.8024	0.8425
	$k = 80$	0.9980	0.9981	0.9763	0.9731	0.9542	0.9500	0.9013	0.9012
	Anisotropic diffusion								
	optimized for UQI	0.7043	0.9827	0.2744	0.9174	0.2538	0.6532	0.1991	0.2939
	optimized for SSIM	0.7043	0.9827	0.2557	0.9559	0.2171	0.9084	0.0972	0.7278

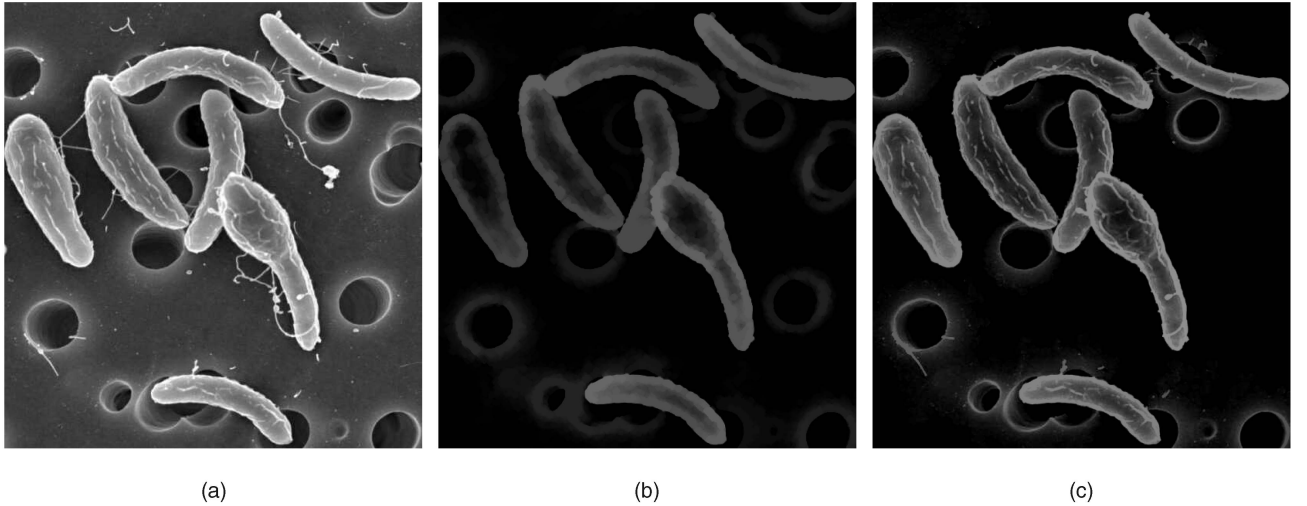


Fig. 10. (a) A group of bacteria linked by fimbriae, (b) the output of the noncompactness filter with $\lambda = 2, k = 0$ and using contraction-based connectivity as a base connectivity class, and (c) the output of the same filter with k set to 80.

whether UQI or SSIM was used. In particular, to improve SSIM, λ had to be decreased to 0.015, and the number of iterations had to be increased. This is unsurprising as the S/N ratio is about 2.

Given this severe corruption, it is the *k*-absorption method that performs very well, as can be verified in Fig. 9. Some more care was needed in setting k and, in some cases, $k = 60$ gave better results due to the lower contrast of the foreground features. A modest improvement in both UQI and SSIM is obtained by postprocessing an area opening with an area threshold of 5, which removes small bright features inside the letters (see Fig. 9e).

Two other anisotropic diffusion methods were tested [33], [43], but performed worse than the Perona-Malik method (data not shown).

4.2 Microscopy

This experiment aims to show the utility of *k*-flat filtering combined with contraction-based connectivity. Key problems are the thin filaments (fimbriae) linking the bacteria in Fig. 10a, which must be removed. These thin, elongated image structures are responsible for what is known as the *leakage problem*, common to all connected operators [27], [44]. Contraction-based second-generation connectivity [12], [23], [29] can, in part, resolve this issue by interpreting pixels in such regions as singleton sets. An area opening configured with this connectivity removes such regions completely for any $\lambda > 1$. This causes severe loss of detail due to oversegmentation [45]. Fig. 10b shows the output of the noncompactness filter ($\lambda = 2, k = 0$) configured with a

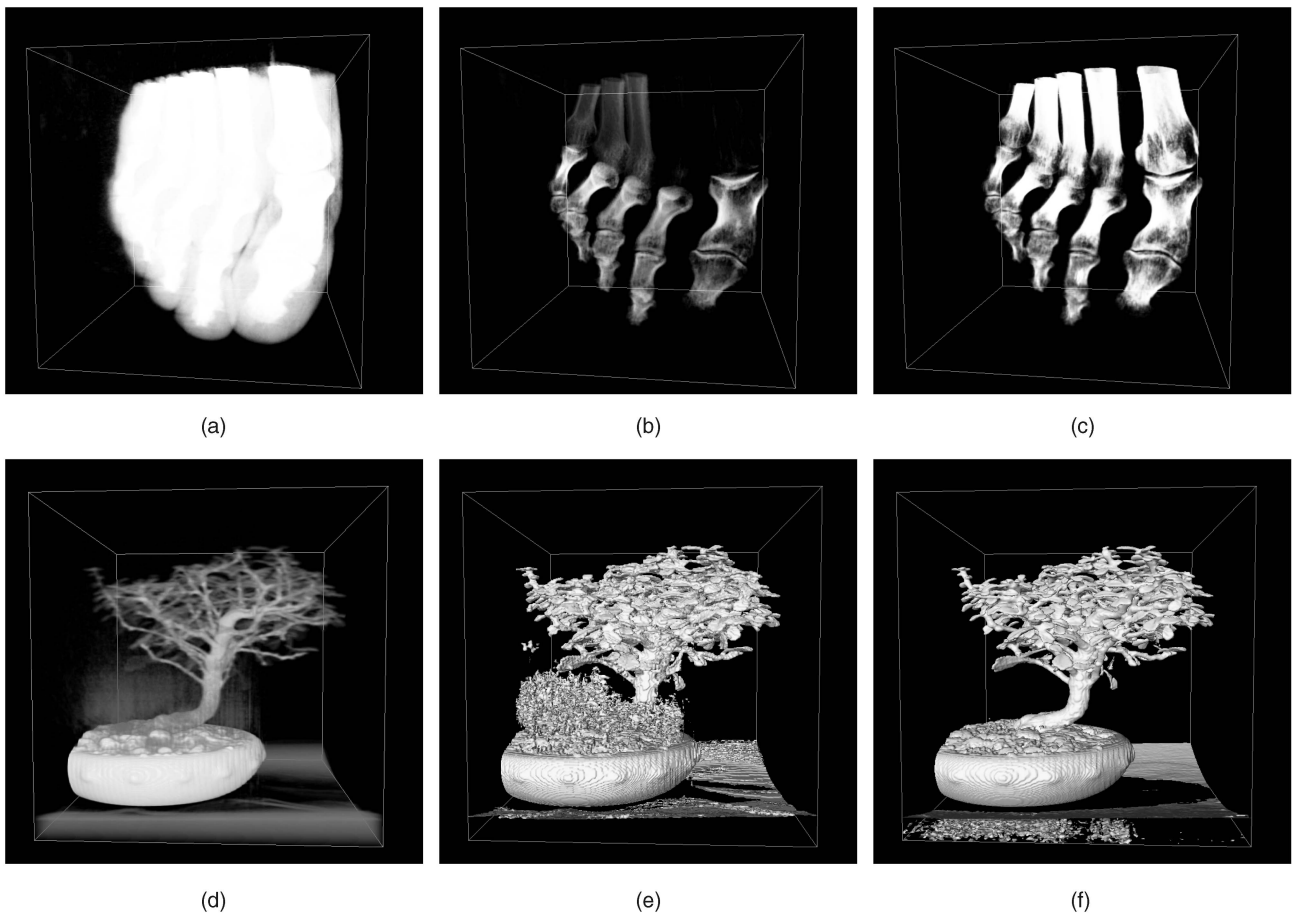


Fig. 11. Experiments with 3D data. Top row: (a) a CT scan of a foot, shown in DVR; (b) the result of the noncompactness filter for $\lambda = 1.2$ and $k = 0$; (c) same as (b) but with $k = 120$. Bottom row: (d) a CT scan of bonsai tree shown in DVR; (e) the result of the sparseness filter for $\lambda = 3.9$ and $k = 0$; and (f) same as (e) but $\lambda = 4.8$ and $k = 40$.

contractive mask-based second-generation connectivity [7] using an opening by a ball of radius 4 as mask. All of the fine details on the surface of the bacteria are lost. Using the same attribute filter with $k = 80$ restores most fine details found within the boundaries of the bacteria (high contrast regions) while it ignores all others, see Fig. 10c.

4.3 CT-Scans

The next two experiments were carried out on 3D data sets aiming at noise reduction and object restoration. Both 8-bit isotropic volumes are of size 256^3 .

The first data set shown in Fig. 11a using direct volume rendering (DVR) is a rotational C-arm X-ray scan of a human foot, courtesy of Philips Research, Hamburg, Germany. The objective is to enhance the bone by suppressing the tissue. Using the 3D extension of the noncompactness filter [11], most of the tissue is suppressed for $\lambda = 1.2$ —Fig. 11b. It can be seen though that the filter fails to retain the integrity of the bone and parts of it, like the upper half of the first two toes, are rejected, leaving only a few elongated components that satisfy the criterion. Moreover, in isosurface projection, not shown here, low-contrast elongated tissue fragments resting on the background are observed. Increasing λ further to eliminate them removes further parts of the bone irreversibly. Instead, these low-contrast tissue fragments can be removed with a

relatively low value for k . Increasing k further ($k = 120$) also allows the recovery of the missing bone, as shown in Fig. 11c.

The second data set, shown in Fig. 11d, is a CT scan of a bonsai tree, courtesy of S. Roettger, VIS, University of Stuttgart. The objective in this case is to remove a cloud of noise surrounding the tree without distorting fine details at the top levels like the leaves. The size and noncompactness filters fail since much of the noise is connected, making it a high volume and certainly a noncompact set. Employing the sparseness attribute from [11], noise is progressively removed until $\lambda = 3.9$ —Fig. 11e. For any $\lambda > 3.9$, the filter, together with noise, distorts the bonsai heavily unless k is adjusted. Since the noise cloud satisfies the criterion but is of lower contrast than the tree, setting k to a value above the maximum noise intensity removes it entirely. Moreover, with $k > 0$, λ can be further increased without losing structures from the bonsai. An optimal result was achieved for $\lambda = 4.8$ and $k = 40$ —Fig. 11f. These last two images are shown in isosurface projection (isolevel 10) to verify the integrity of fine details like the leaves.

Computation times for both data sets are listed in Table 2. All experiments were carried out on an Intel Core 2 Duo E8400 at 3.0 GHz with 2 GB memory, using only a single core. As can be seen, the difference between using the k -subtractive filter with $k = 0$ and simply using the standard subtractive rule is between 1 and 6 ms on these data, a value which pales

TABLE 2
CPU Timings for the 16 MB 3D Data Sets (in Seconds)

Data Set	Attribute filter timings				k
	build	standard filter	k -flat filter		
			$k = 0$	$k > 0$	
CT Foot	7.890	0.0148	0.0207	0.0205	120
CT Tree	6.820	0.0112	0.0124	0.0125	40

with the rendering time of between 50 and 250 ms needed to show the results (depending largely on λ). Changing k and rendering the result can be done at a rate of between 3.5 and 18 frames per second, i.e., fully interactively.

4.4 Variable k Filtering

Until now, we have assumed that k is constant, implying that *unimportant* variations in brightness do not correlate with intensity. This holds for uniform Gaussian noise or roughly constant texture. In this case, we can select k using available knowledge of the σ of the Gaussian noise or the known texture properties.

In general, we may want to replace the constant k in Algorithm 1 by some increasing function $k(h)$, mapping gray-level k -value. This can be used in instances when we know that correlations with intensity exist. One example is the case of photon-noise-limited images prevalent in astronomy, where the Poisson noise is proportional to the square root of the signal. We obtain

$$k(h) = \kappa_{\frac{1}{2}} \sqrt{h}. \quad (35)$$

If h is expressed in photon counts, $\kappa_{\frac{1}{2}}$ determines how many standard deviations are needed for a fluctuation to be significant, giving statistical meaning to $\kappa_{\frac{1}{2}}$. Of course, a more general form is

$$k(h) = \kappa_p h^p, \quad (36)$$

to cater to any power law relationship expected between gray level and expected fluctuation amplitude. For the moment, we will focus on $p = 1$, making it a linear function. In this way, κ_1 determines the *relative* intensity fluctuations, which are considered unimportant rather than absolute. This is more similar to the human visual system in the way it deals with image contrast and yields invariance to linear scaling in gray level. The filter then becomes invariant to the exposure time, telescope aperture, and sensitivity of the imaging device.

To implement this, all we need to change in Algorithm 1 is to pass a function parameter $kFunc$, instead of k , and declare a local variable k instead, and add the line

$$(L1) \quad k = kFunc(node.Level);$$

before the use of k when processing each node. The price we pay for this is loss of idempotence in the case of nonincreasing criteria. If the criterion used is increasing, each node that is preserved because it passes the criterion is preserved *at its original gray level*. Therefore, in a second filtering pass, the same value of k will be used for them, yielding the same result. This is not true for nonincreasing attributes in the subtractive filtering rule. If, e.g., the root

node (background) is lowered in gray level because it fails the criterion, all retained nodes nested within are lowered by at least as much. Therefore, in a second filtering pass, they will receive a lower k value, and their descendents will, in general, not be preserved in the same way.

In the case of Poisson noise, using (35), it really only makes sense to use the *current* gray level h of a node to determine k because the noise level is determined by that gray level. However, if we use $p = 1$ in (36), we could use a different strategy. In Algorithm 1, we observe that gray levels are assigned from the root toward the leaves, i.e., from low gray levels to high. For the root node, we use $k(h)$ as before, with h the current gray level. For every other node we visit, the parent has already been assigned a new gray level. We can therefore compute the new gray level \tilde{h} , assuming that the node is preserved because it passes the criterion. If we use $k(\tilde{h})$ as k -value in Algorithm 1, this guarantees that the same k value will be used in the next application of the filter, *if* it is preserved, because it passes the criterion. If it does not pass the criterion, its k value becomes irrelevant. To implement this, we use the following line of code:

$$(L2) \quad k = kFunc(parent.NewLevel + difflevel);$$

instead of (L1).

Results of this approach on an image of M81 are shown in Fig. 12. As can be seen in Figs. 12a and 12b, using fixed $k = 20$ preserves more detail than the area attribute filter in Fig. 1d, but also boosts the stellar component. Using the linear form $k = 0.3125h$ yields much better results, both in terms of preservation of galactic detail and in suppression of stars (Figs. 12c and 12d). Using $k = 0.5\tilde{h}$, which, unlike the previous method, is idempotent, yields the best results (Figs. 12e and 12f).

5 DISCUSSION

The notion of k -flat zones can easily be extended to vector images, using some metric in the vector space to limit the maximum distance in color space, similarly to [28]. However, attribute filters as defined in [4] require a total ordering on the values of the pixels, so the algorithm cannot be extended trivially to color and other vector images. It is possible that the work of Gimenez and Evans [46] could be extended in a k -flat sense, greatly increasing the range of applications for k -flat filtering.

Tuning the parameter k depends on the input image, just like with λ . To remove low contrast but high attribute measure details in the background, small values are usually sufficient. If these structures are within larger ones that should also be removed, together with an appropriate value for λ , a high enough value for k is required that exceeds the maximum span of gray levels of the targeted objects. This must ensure that for any component which satisfies the filter criterion, there is not enough contrast to define a k -peak component. To recover components that would be removed by a filter at $k = 0$, a high enough value for k is required such that there exist k -peak components that satisfy the criterion less than k levels below them. For $k = 0$, we simply revert to connected attribute filters.

As in any other filter, knowledge of the standard deviation of noise or amplitude of any unwanted structure

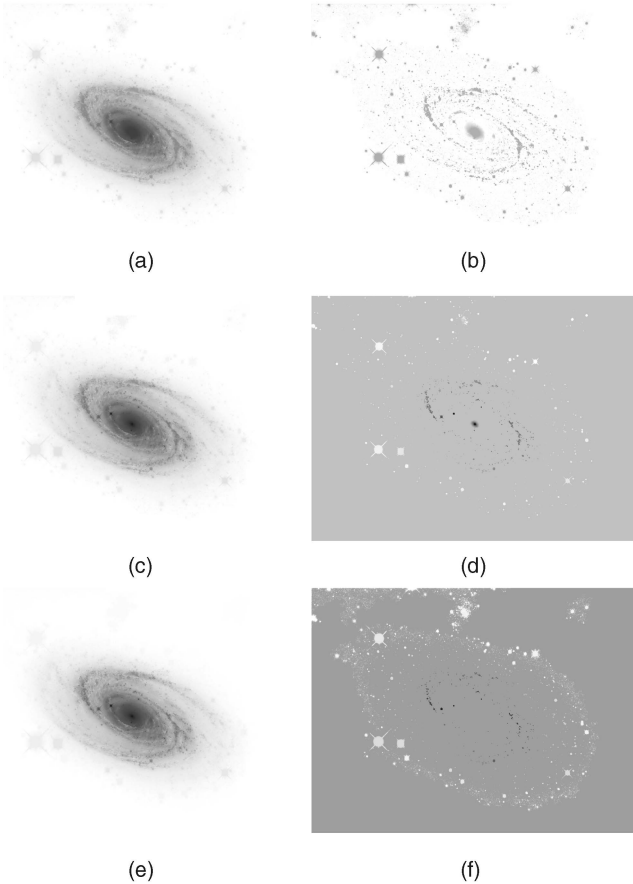


Fig. 12. Filtering results for M81 as in Fig. 1a using the k -flat method: (a) same area filter as in Fig. 1d, but with $k = 20$; (b) contrast-stretched difference between $k = 0$ and $k = 20$ showing better enhancement of galaxy, but increased signal from stars; (c) the same as (a) but with $k = 0.3125h$, using original image brightness; (d) the difference between (c) and (a); (e) the same as (c) but with $k = 0.5h$, using filtered gray level to modify k ; (f) the difference between (e) and (c). In (d) and (f), the gray background indicates the zero level, lighter areas show increased suppression of stellar signal in (c) and (e), respectively; the darker structures show increased galactic detail.

can be used to set k sensibly. Using the variable- k approach it is even possible to add knowledge of scaling laws or correlations of noise to gray levels.

In document processing, it was quite easy to tune k . Unlike the case of anisotropic diffusion, only occasional changes in k were needed for optimality, and the performance degrades only very gently off-optimum. In this application, the shapes and sizes of the image artifacts are similar to the desired objects. It is only contrast which separates the two. This is why the k -absorption filter in particular outperforms its competitors. It achieves this high level of performance even at an S/N ratio of 2, at about 1 percent of the processing time of the anisotropic diffusion filters.

The results obtained in the galactic images are encouraging, though by no means perfect. In the future, we aim to add automatic image inpainting, as developed by [47], to deal with the residues left by extremely bright stars. Better scale-invariant attribute will need to be combined with the k -flat method for full automation.

The computational complexity of the building phase of the Max-Tree algorithm is discussed in various papers [39],

[48], and dominates the computing time of the filtering process, as shown in Table 2. Once the tree structure is built, the *SetPeaks()* function is called if $k > 0$. This is a single pass through the tree and contributes a fixed, but very small time overhead. Similarly, the k -subtractive and k -absorption rules require one pass of the tree involving a few additional *if* – *else* statements. This depends on the number of nodes and is independent of both λ and k . For *SetPeaks()* and the filtering stage, we have a worst-case complexity of $O(N)$ because no more Max-Tree nodes than pixels can exist.

6 CONCLUSIONS

In this paper, we presented hyperconnected attribute filters based on k -flat zones. We gave a general proof that the main properties of connected attribute filters carry over to the hyperconnected case. The aim was to improve retention of detail in objects detected by the filter, while simultaneously rejecting low-contrast background detail. The strength of this effect is controlled by single additional parameter k . The benefits of this demonstrated by our experiments on 2D and 3D data.

Hyperconnectivity [3], [49] is a recent introduction in the theory of connectivity and there are open challenges in both the theoretical and practical sides of it. Most important are the axiomatics of the families of operators which return hyperconnected components marked by points $x \in E$. We bypassed this limitation by using easily detectable hyperconnected components: the k -flat zones. This approach cannot be generalized and the problem remains to be solved. Note that there are no restrictions on the nature of this base connectivity class, meaning that second generation and other derivatives of standard connectivity may also be used. This was demonstrated on the second experiment of Section 4. The algorithm for hyperconnected attribute filters can be used for classical attribute filters by setting $k = 0$, and can be combined with second-generation connectivities.

Hyperconnected attribute filters were implemented on the Max-Tree structure in the form of a filtering rule. This allows setting k interactively, which is of great benefit in 3D. A demo program, sample data sets, and source code are available at <http://www.cs.rug.nl/~michael/MTdemo/>. This program contains implementations of k -flat versions of other filtering rules. The CPU timings reported for each data set show that the overhead of the new method deviates less than 1 percent from that of regular connected attribute filters and the algorithm is linear with image size.

In future work, we are looking into the axiomatics of hyperconnectivity and hyperconnected components. This is an essential step for exploring the field of hyperconnected morphology and its operators.

ACKNOWLEDGMENTS

The authors would like to thank Scott Trager of the Kapteyn Astronomical Institute, University of Groningen for his input on galaxy star separation. This work is funded by the Netherlands Organization for Scientific Research under project number 612.065.202.

REFERENCES

- [1] P. Salembier and J. Serra, "Flat Zones Filtering, Connected Operators, and Filters by Reconstruction," *IEEE Trans. Image Processing*, vol. 4, no. 8, pp. 1153-1160, Aug. 1995.
- [2] J. Serra, "Connectivity on Complete Lattices," *J. Math. Imaging and Vision*, vol. 9, pp. 231-251, 1998.
- [3] U. Braga-Neto and J. Goutsias, "A Theoretical Tour of Connectivity in Image Processing and Analysis," *J. Math. Imaging and Vision*, vol. 19, pp. 5-31, 2003.
- [4] E.J. Breen and R. Jones, "Attribute Openings, Thinnings and Granulometries," *Computer Vision and Image Understanding*, vol. 64, no. 3, pp. 377-389, 1996.
- [5] L. Vincent, "Morphological Grayscale Reconstruction in Image Analysis: Application and Efficient Algorithm," *IEEE Trans. Image Processing*, vol. 2, no. 2, pp. 176-201, Apr. 1993.
- [6] M.H.F. Wilkinson and M.A. Westenberg, "Shape Preserving Filament Enhancement Filtering," *Proc. Int'l Conf. Medical Image Computing and Computer-Assisted Intervention*, W.J. Niessen and M.A. Viergever, eds., pp. 770-777, 2001.
- [7] G.K. Ouzounis and M.H.F. Wilkinson, "Mask-Based Second-Generation Connectivity and Attribute Filters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 990-1004, June 2007.
- [8] D. Gatica-Perez, C. Gu, M.T. Sun, and S. Ruiz-Correa, "Extensive Partition Operators, Gray-Level Connected Operators, and Region Merging/Classification Segmentation Algorithms: Theoretical Links," *IEEE Trans. Image Processing*, vol. 10, no. 9, pp. 1332-1345, Sept. 2001.
- [9] J. Crespo, R.W. Schafer, J. Serra, C. Gratin, and F. Meyer, "The Flat Zone Approach: A General Low-Level Region Merging Segmentation Method," *Signal Processing*, vol. 62, pp. 37-60, 1997.
- [10] R. Jones, "Connected Filtering and Segmentation Using Component Trees," *Computer Vision and Image Understanding*, vol. 75, pp. 215-228, 1999.
- [11] M.A. Westenberg, J.B.T.M. Roerdink, and M.H.F. Wilkinson, "Volumetric Attribute Filtering and Interactive Visualization Using the Max-Tree Representation," *IEEE Trans. Image Processing*, vol. 16, no. 12, pp. 2943-2952, Dec. 2007.
- [12] *Image Analysis and Mathematical Morphology. II: Theoretical Advances*, J. Serra, ed. Academic Press, 1988.
- [13] G. Matheron and J. Serra, "Strong Filters Connectivity," *Image Analysis and Math. Morphology*, J. Serra, ed., vol. 2, pp. 141-157, Academic Press, 1988.
- [14] H.J.A.M. Heijmans, "Morphological Filters," *Proc. Summer School on Morphological Image and Signal Processing*, 1995.
- [15] J. Crespo, J. Serra, and R.W. Schafer, "Connected Morphological Operators for Binary Images," *Computer Vision and Image Understanding*, vol. 73, pp. 99-120, 1999.
- [16] J. Crespo, J. Serra, and R.W. Schafer, "Theoretical Aspects of Morphological Filters by Reconstruction," *Signal Processing*, vol. 47, pp. 201-225, 1995.
- [17] J. Crespo and R.W. Schafer, "Locality and Adjacency Stability Constraints for Morphological Connected Operators," *J. Math. Imaging and Vision*, vol. 7, pp. 85-102, 1997.
- [18] V. Caselles and P. Monasse, "Grain Filters," *J. Math. Imaging and Vision*, vol. 17, pp. 249-270, 2002.
- [19] J.A. Moore, K.A. Pimblett, and M.J. Drinkwater, "Mathematical Morphology: Star/Galaxy Differentiation & Galaxy Morphology Classification," *Publications of the Astronomical Soc. of Australia*, vol. 23, pp. 135-146, 2006.
- [20] Z. Frei, P. Guhathakurta, J. Gunn, and J. Tyson, "A Catalog of Digital Images of 113 Nearby Galaxies," *The Astronomical J.*, vol. 111, pp. 174-181, 1996.
- [21] Z. Frei, "Semi-Automatic Removal of Foreground Stars from Images of Galaxies," *Publications of the Astronomical Soc. of the Pacific*, vol. 108, pp. 624-632, 1996.
- [22] S. Kendall, R.C. Kennicutt, C. Clarke, and M.D. Thornley, "Tracing Spiral Density Waves in M81," *Monthly Notices of the Royal Astronomical Soc.*, vol. 387, pp. 1007-1020, 2008.
- [23] C. Ronse, "Set-Theoretical Algebraic Approaches to Connectivity in Continuous or Digital Spaces," *J. Math. Imaging and Vision*, vol. 8, pp. 41-58, 1998.
- [24] J. Serra, "Connections for Sets and Functions," *Fundamenta Informaticae*, vol. 41, pp. 147-186, 2000.
- [25] M. Nagao, T. Matsuyama, and Y. Ikeda, "Region Extraction and Shape Analysis in Aerial Photographs," *Computer Graphics and Image Processing*, vol. 10, no. 3, pp. 195-223, 1979.
- [26] F. Meyer, "The Levelings," *Proc. Fourth Int'l Symp. Math. Morphology*, pp. 199-206, June 1998.
- [27] P. Salembier, A. Oliveras, and L. Garrido, "Anti-Extensive Connected Operators for Image and Sequence Processing," *IEEE Trans. Image Processing*, vol. 7, no. 4, pp. 555-570, Apr. 1998.
- [28] P. Soille, "Constrained Connectivity for Hierarchical Image Decomposition and Simplification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1132-1145, July 2008.
- [29] U. Braga-Neto and J. Goutsias, "Connectivity on Complete Lattices: New Results," *Computer Vision and Image Understanding*, vol. 85, pp. 22-53, 2002.
- [30] B. Naegel, N. Passat, N. Boch, and M. Kocher, "Segmentation Using Vector-Attribute Filters: Methodology and Application to Dermatological Imaging," *Proc. Int'l Symp. Math. Morphology*, pp. 239-250, 2007.
- [31] K.E. Purnama, M.H.F. Wilkinson, A.G. Veldhuizen, P.M.A. van Ooijen, J. Lubbers, T.A. Sardjono, and G.J. Verkerke, "Branches Filtering Approach for Max-Tree," *Proc. Second Int'l Conf. Computer Vision Theory and Applications*, pp. 328-332, Mar. 2007.
- [32] P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629-639, July 1990.
- [33] J. Weickert, "Coherence-Enhancing Diffusion Filtering," *Int'l J. Computer Vision*, vol. 31, nos. 2/3, pp. 111-127, 1999.
- [34] L. Vincent, "Morphological Area Openings and Closings for Grayscale Images," *Proc. NATO Shape in Picture Workshop*, pp. 197-208, Sept. 1992.
- [35] F. Cheng and A.N. Venetsanopoulos, "An Adaptive Morphological Filter for Image Processing," *IEEE Trans. Image Processing*, vol. 1, no. 4, pp. 533-539, Oct. 1992.
- [36] P. Maragos and R.D. Ziff, "Threshold Superposition in Morphological Image Analysis Systems," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 498-504, May 1990.
- [37] E.R. Urbach, J.B.T.M. Roerdink, and M.H.F. Wilkinson, "Connected Shape-Size Pattern Spectra for Rotation and Scale-Invariant Classification of Gray-Scale Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 272-285, Feb. 2007.
- [38] U. Braga-Neto and J. Goutsias, "Grayscale Level Connectivity: Theory and Applications," *IEEE Trans. Image Processing*, vol. 13, no. 12, pp. 1567-1580, Dec. 2004.
- [39] L. Najman and M. Couprie, "Building the Component Tree in Quasi-Linear Time," *IEEE Trans. Image Processing*, vol. 15, no. 11, pp. 3531-3539, Nov. 2006.
- [40] M.H.F. Wilkinson, H. Gao, W.H. Hesselink, J.E. Jonker, and A. Meijster, "Concurrent Computation of Attribute Filters Using Shared Memory Parallel Machines," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1800-1813, Oct. 2008.
- [41] Z. Wang and A. Bovik, "A Universal Image Quality Index," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81-84, Mar. 2002.
- [42] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600-612, Apr. 2004.
- [43] Y. Yu and S. Acton, "Speckle Reducing Anisotropic Diffusion," *IEEE Trans. Image Processing*, vol. 11, no. 11, pp. 1260-1270, Nov. 2002.
- [44] C. Tzafestas and P. Maragos, "Shape Connectivity: Multiscale Analysis and Application to Generalized Granulometries," *J. Math. Imaging and Vision*, vol. 17, pp. 109-129, 2002.
- [45] G.K. Ouzounis and M.H.F. Wilkinson, "Countering Oversegmentation in Partitioning-Based Connectivities," *Proc. Int'l Conf. Image Processing*, pp. 844-847, 2005.
- [46] D. Gimenez and A.N. Evans, "An Evaluation of Area Morphology Scale-Spaces for Colour Images," *Computer Vision and Image Understanding*, vol. 110, no. 1, pp. 32-42, 2008.
- [47] M. Dimiccoli and P. Salembier, "Perceptual Filtering with Connected Operators and Image Inpainting," *Proc. Int'l Symp. Math. Morphology*, pp. 227-238, 2007.
- [48] A. Meijster and M.H.F. Wilkinson, "A Comparison of Algorithms for Connected Set Openings and Closings," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 484-494, Apr. 2002.
- [49] J. Serra, "A Lattice Approach to Image Segmentation," *J. Math. Imaging and Vision*, vol. 24, pp. 83-130, 2006.



Georgios K. Ouzounis received the MPhil degree in computer science from the Department of Computing and Electrical Engineering, Heriot Watt University, Edinburgh, Scotland, in 2001, and the PhD degree in image analysis and connected morphological filters from the Institute of Mathematics and Computing Science, University of Groningen, The Netherlands, in 2009. From January 2008 to September 2009, he worked at the

Democritus University of Thrace—2nd Department of Surgery, University General Hospital of Alexandroupoli, Greece, as a medical image analyst, developing 3D CT segmentation methods for diagnostic purposes of interest to the Urological Clinic. Currently, he is a researcher at the Global Security and Crisis Management Unit of ISFEREA—Joint Research Center of the European Commission in Ispra, Italy, where he works on the development of advanced VHR satellite image analysis algorithms.



Michael H.F. Wilkinson received the MSc degree in astronomy from the Kapteyn Laboratory, University of Groningen in 1993, and the PhD degree from the Institute of Mathematics and Computing Science, University of Groningen, in 1995, where he worked on image analysis of intestinal bacteria in the Department of Medical Microbiology. He was appointed a researcher at the Centre for High-Performance Computing in Groningen working

on simulating the intestinal microbial ecosystem on parallel computers. During that time, he edited the book *Digital Image Analysis of Microbes* (John Wiley, UK, 1998) together with Frits Schut. After this, he worked as a researcher at the Johann Bernoulli Institute for Mathematics and Computer Science (JBI) on image analysis of diatoms. He is currently a senior lecturer at JBI working on morphological image analysis and, especially, connected morphology. He is a senior member of the IEEE.

► **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.**